



UNIVERSITAT DE
BARCELONA

Treball final de grau
GRAU D'ENGINYERIA
INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Colorimetric analysis for biostrips using
Python and OpenCV

Autor: Alicia Morales Carrasco

Directors: Dr. J.Daniel Prades

Dr. Eloi Puertas

Realitzat a: Departament d'Enginyeria Electrònica
i Biomèdica

Departament Matemàtiques i Informàtica

Barcelona, 27 de juny de 2018

Abstract

The dipstick tests provide a quick means to carry out the chemical analysis of liquid poured on a strip. In this work we study and propose an application of colorimetric analysis of the reactive strips in order to determine quantitatively the test results through the processing and computer analysis of reactive strip images. It explains the ingenuity made to locate the strip in the image, detect the area of interest, make the color correction using a laboratory of colors surveyed and the colorimetric analysis of the results. Finally, the precision of the method is evaluated by the creation and analysis of strip simulations.

Resum

Les tires reactives proporcionen un mitjà ràpid per dur a terme l'anàlisi química d'un líquid vessat a una tira. En aquest treball s'estudia i proposa una aplicació d'anàlisi colorimètric de les tires reactives per tal de determinar quantitativament el resultat del test realitzat amb les tires, mitjançant el processat i anàlisi per computador d'imatges de tires reactives. S'explica l'enginy realitzat per localitzar la tira en la imatge, detectar la zona d'interès, realitzar la correcció de color mitjançant l'ús d'un laboratori de colors enquestat a la imatge i l'anàlisi colorimètric dels resultats. Finalment, s'avalua la precisió del mètode mitjançant la creació i anàlisi de simulacions de tires.

Agraïments

Vull agrair als meus tutors Dr. J. Daniel Prades i Dr. Eloi Puertas la seva paciència i dedicació en la tutorització d'aquest treball. També agrair a l'equip de desenvolupadors de *ColorSensing* que sempre han resolt amablement els meus dubtes sobre els mòduls de correcció de color i detecció de QRs utilitzats en aquest treball.

Índex

1	Introducció	1
1.1	Context	1
1.2	Motivació	1
1.3	Objectius	1
1.4	Estructura de la memòria	2
2	Conceptes previs	3
2.1	Sobre les tires reactives	3
2.2	Sobre els Colored Quick Response Codes (QRs)	5
2.3	Orientació del QR i correcció de la perspectiva	8
3	Anàlisi	11
3.1	La detecció de la tira de test a l'espai de la imatge	11
3.2	La correcció de color de la imatge	11
3.3	La quantificació de color	11
4	Disseny	12
4.1	El prototip on s'introduiran les tires	12
4.2	El posicionament dels QRs al prototip	13
4.3	Creació de QRs amb color	13
5	Desenvolupament	15
5.1	Correcció de color	16
5.2	Detecció del prototip i zona d'interès	19
5.3	Quantificació de color	24

6	Planificació	27
7	Proves i resultats	29
7.1	Creació de tires de test artificials	29
7.2	Anàlisi dels resultats	29
8	Conclusions i treball futur	34

1 Introducció

1.1 Context

L'ús de tires reactives és prou conegut en test d'embaràs i d'orina, proporcionen un mitjà ràpid per dur a terme l'anàlisi química d'un líquid vessat a una tira. L'avaluació dels resultats del test es basa en la comparació dels colors a ull nu de la tira amb els de l'escala cromàtica proveïda pel fabricant. En aquest treball es proposa una aplicació d'anàlisi colorimètric de les tires reactives per tal de determinar quantitativament la concentració de cadascun dels compostos del líquid mitjançant el processat i l'anàlisi per computador d'imatges de tires reactives.

1.2 Motivació

Encara que les tires reactives proporcionen prou informació qualitativa, l'anàlisi intensiu del fluid requereix mètodes amb maquinària especialitzada. La motivació del treball és poder arribar a construir una metodologia d'anàlisi més econòmica, portable i propera a l'usuari.

Les eines i coneixements adquirits al Grau d'Enginyeria Informàtica no només ens possibilitaran la implementació de la part tècnica, sinó que ens proporcionaran una visió crítica i proactiva en la gestió i desenvolupament del projecte.

1.3 Objectius

- Utilitzar els coneixements bàsics i mètodes de reconeixement i processament d'imatges que han estat treballats a assignatures del grau per tal de detectar la tira reactiva.
- Reconèixer l'àrea d'interès a tractar a la imatge o (ROI), la regió dels resultats de la tira, mitjançant mètodes de visió per computador i dos QRs.
- Corregir el color de la imatge amb un laboratori de colors inserits dins els QRs.

- Determinar qualitativament la quantitat de color de la tira de test.

Treballaré paral·lelament amb l'equip *ColorSensing* de la Facultat de Física de la Universitat de Barcelona que està desenvolupant uns mòduls de detecció de QRs i correcció de color i ens han proposat fer-los servir per a la nostra tasca.

1.4 Estructura de la memòria

Al treball veurem com portar a la pràctica la detecció de la tira de test i la quantificació del color. A la memòria començarem per parlar de l'anàlisi del problema, la divisió del problema en parts i els diferents mètodes que creiem que ens podrien fer servei per a cada tasca. Veurem com s'ha planificat de manera iterativa i unitària cadascuna de les divisions del problema, a continuació explicarem l'enginy realitzat per resoldre els problemes individuals de l'anàlisi i seguidament examinarem els detalls de la implementació de cadascuna de les parts.

2 Conceptes previs

En serà d'utilitat conèixer la natura de la tira, el funcionament i estructura de QRs i espais de color.

2.1 Sobre les tires reactives

Les tires reactives o biostrips solen fabricar-se de paper o cartró i s'impregnen amb reactius que indiquen alguna característica del líquid vessat a la tira canviant de color. En medicina s'usen com a eina de diagnosi per detectar la presència d'una substància determinada, els anàlits. Per exemple, les tires biomètriques d'orina s'utilitzen per detectar a l'orina l'hemoglobina, els nitrits produïts per bacteris en una infecció urinària, proteïnes, nitrocel·lulosa, glucosa, urobilinògens o cetones.

Les tires reactives més habituals són com les de la figura 1, contenen dues ratlles de substàncies (anti-mouse i anti-analyte) i quan es vessa el líquid, l'anti-mouse sempre s'activa però l'anti-analyte només si l'anàlit és present a la substància (resultat positiu). La intensitat del color que desprengui l'anti-analyte serà proporcional a la quantitat de presència de l'anàlit al líquid.

Treballarem sobre tires reactives de paper de 7 x 0.5 cm. Com a resultat de l'anàlisi de la tira podríem mostrar la diferència d'intensitat de color de la zona de l'anti-analyt respecte a la zona de l'anti-mouse i/o com a valor RGB.

El fabricant proporciona per cadascuna de les tires una paleta de colors i el significat que representa.

2.1.1 Sobre els models de pic que podem trobar a les tires

Els fenòmens d'absorció presents a la tira engendraran un resultat en forma de pic a la zona d'interès de la tira (la zona de resultat). A l'hora de quantificar el color haurem d'ajustar el resultat als model de pic més usuals presentats en aquest tipus de fenomen.

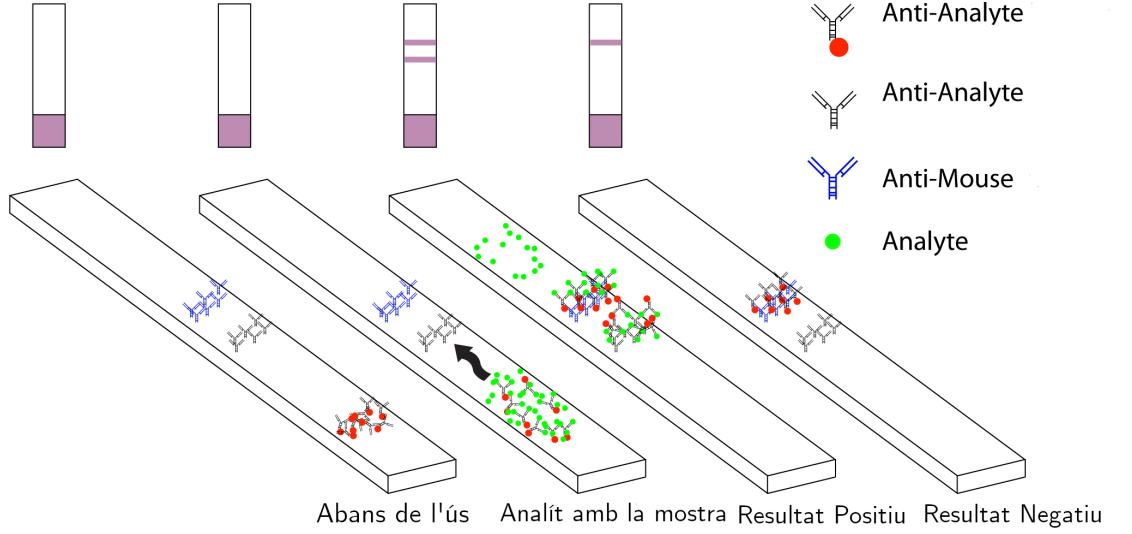


Figura 1: Tires reactives, funcionament i resultats possibles

Model Gaussià

$$f(x; A, \mu, \sigma) = \frac{A}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

On el paràmetre A és l' amplitud, μ és la posició del centre del pic i σ (la desviació mitjana) l'amplada de la campana. Veure a la Figura 2 en blau.

Model Loretzia

$$f(x; A, \mu, \sigma) = \frac{A}{\pi} \left[\frac{\sigma}{(x - \mu)^2 + \sigma^2} \right]$$

On el paràmetre A is és l'amplitud, μ és la posició del centre del pic i 2σ és l'amplada a la meitat de l'amplitud. Veuren a la Figura 2 en taronja.

Model Pseudo-Voigt

$$f(x; A, \mu, \sigma, \alpha) = \frac{(1 - \alpha)A}{\sigma_g\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma_g^2} + \frac{\alpha A}{\pi} \left[\frac{\sigma}{(x - \mu)^2 + \sigma^2} \right]$$

Que representa la suma ponderada d'una gaussiana i una lorentziana que comparteixen els valors d'amplitud A centre (μ) i amplada màxima a meitat de l'amplitud. El paràmetre α controla els pes relatiu que es dona a les funcions gaussiana i lorentziana respectivament. Veure a la Figura 2 en verd.

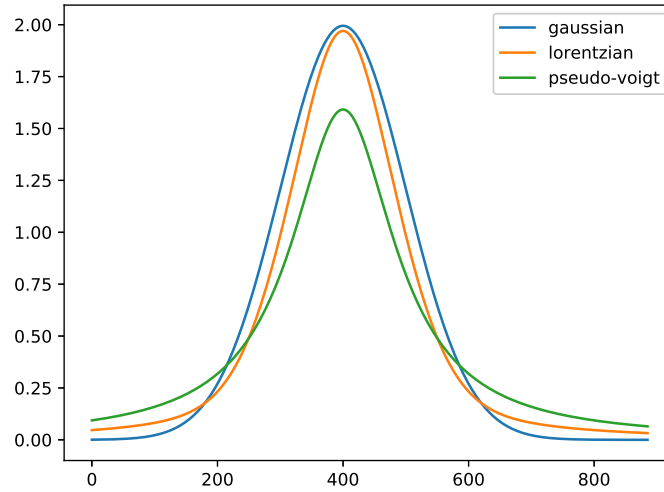


Figura 2: Models de pics amb $\mu = 400$, $\sigma = 200$, amplitud = 2, 1.5 i 1.97 respectivament i $\alpha = 0.5$ a la pseudo-voigt

2.2 Sobre els Colored Quick Response Codes (QRs)

Els codis QR van ser desenvolupats en 1994 per Denso-Wave, una companyia de Toyota. Les ISO estableixen com codificar i decodificar la informació així com les mides de QR existents, també anomenades versions. La mida del prototip condiciona els tipus de QR que es poden incrustar en ell. Es pretén que el mètode sigui portable, cosa que en ens limita a QRs de versió menor o igual a 3. (mida 29x29) Per tal d'inserir els colors dins del QR hem de conèixer l'estructura interna i el procés de detecció i decodificació de la informació d'un QR. [2][3][4] [5]

2.2.1 Parts d'un QR

La matriu del QR és divideix principalment en zones que contenen informació imprescindible per la detecció del QR i decodificació de les dades, com podem veure a la Figura 3, tenim:

1. **La versió:** Que indica la mida del QR o el que és el mateix, el nombre de mòduls.

2. **Format information:** que indica el tipus d'informació s'emmagatzemament dins el QR.
3. **Data and error correction keys:** les dades i altra informació per tal de corregir l'error que es pugui engendrar en la detecció del QR.
4. (a) **Els position patterns:** tres patrons de quadrats amb relació 3:5:7 posicionats a les voreres superior-esquerre, superior-dreta i inferior-esquerre que indiquen l'orientació del QR.
 (b) **Alignment patterns:** tres quadrats concèntrics que permeten corregir orientació per tal de detectar millor les dades. Només els QRs amb versió 2 o superior en tenen.
 (c) **Timing patterns:** mòduls blancs i negres alternant que permeten identificar la densitat del símbol i la versió.
5. **Quiet zone:** Una zona en blanc que envolta el símbol per tal de definir el valor de reflectància nominal amb la qual els mòduls blancs s'han de detectar.

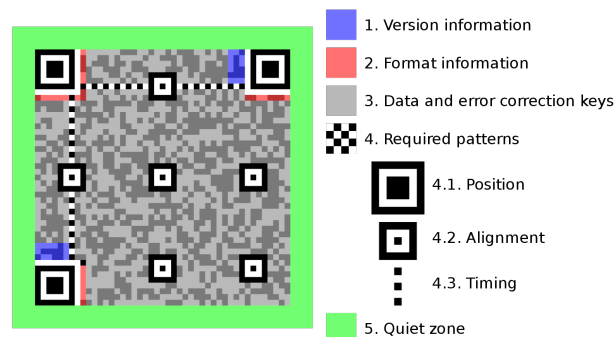


Figura 3: Estructura d'un *Quick Response Code*

2.2.2 Descodificació del QR

El procés de descodificació del QR va des de trobar el símbol a la imatge fins a recuperar la informació que emmagatzema. Els passos a gran escala són:

1. Determinar un llindar global agafant el valor de reflectància mitjana de la imatge.

2. Localitzar els position patterns gràcies a les propietats descrites en la secció secció 2.2.1, 4a. Veure Figura 4.
3. Determinar l'orientació del símbol analitzant els position patterns
4. Determinar la versió i llegir la informació de la versió si es tracta d'una versió 7 o superior.
5. Si la versió té alignment patterns intentar corregir orientació amb ells.
6. Construir una matriu de bits clars-obscurs basats en el llindar definit a 1.
7. Descodificar el format de la informació, el nivell de correcció d'errors i el patró de màscara necessari per descodificar la informació.
8. Recuperar el missatge original com a cadena de bits assemblant els blocs de dades en una única seqüència i subdividint-los en segments que finalment es descodifiquen com a caràcters depenent el tipus de dades.

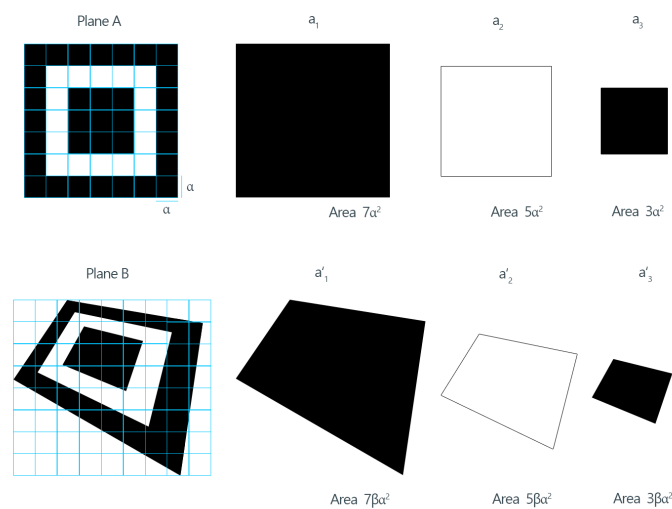


Figura 4: La màgia de la detecció de QRs, la relació única dels position patterns

2.3 Orientació del QR i correcció de la perspectiva

Una vegada s'han trobat els position patterns s'usaran per a conèixer l'orientació i fer la transformació projectiva que ens deixarà el QR de cara i ens permetrà extreure-hi la informació (i els colors). L'algoritme ha de saber quin position pattern correspon a cadascuna de les cantonades i l'orientació d'aquest.

Per fer-ho és tan fàcil com calcular les distàncies entre cada parell de position patterns, els que tinguin distància superior seran els que corresponen als extrems de la diagonal del quadrat. Després per saber a quin extrem de la diagonal pertany cadascun es descobreix amb el signe del pendent respecte l'altre punt que no pertany a la diagonal. Veieu Figura 5.

Per corregir la perspectiva és necessari conèixer els quatre punts de cada canto-

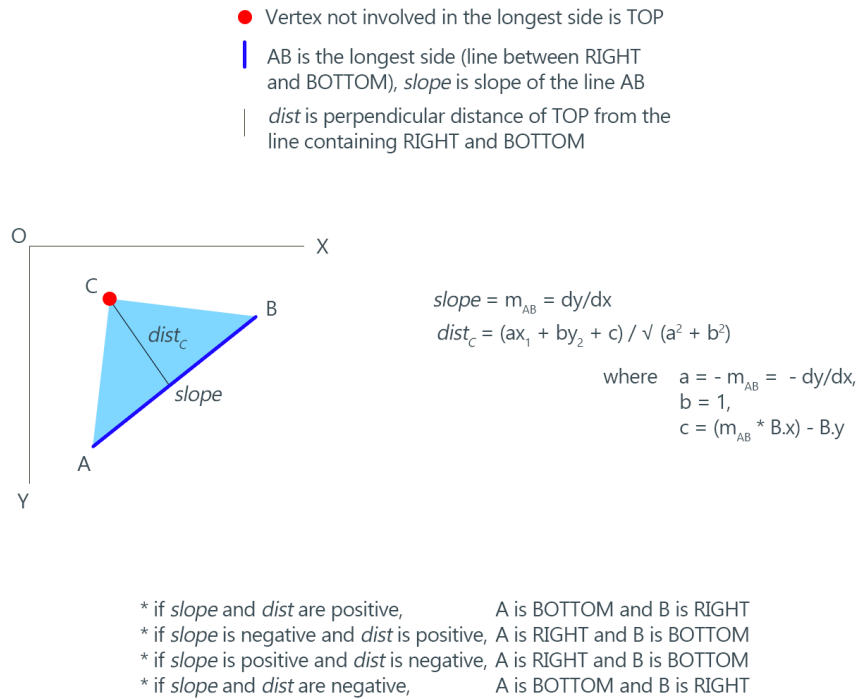


Figura 5: Metodologia per trobar l'orientació d'un QR.

nada. Amb els position patterns en tenim tres, per aconseguir el quart el que es fa aproximar el quart punt com el punt que es troba tirant dues rectes des de els

punts dels extrems.

Amb els quatre punts ja es corregeix la perspectiva però per culpa de l'aproximació del quart punt, és probable que a la zona propera al punt es produeixin errors que impossibilitin la lectura de les dades de dita zona. El que es fa és usar els alignment patterns per tractar de millorar la projecció d'aquesta zona.

2.3.1 Sobre les imatges i els colors

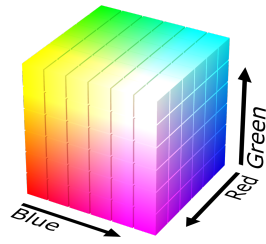
Les característiques utilitzades per distingir un color d'un altre solen ser la lluminositat, la tonalitat i la saturació. La lluminositat ens aporta la noció cromàtica de la intensitat, la tonalitat està relacionada amb la longitud d'ona predominant que reflecteix i la saturació es refereix amb la quantitat de llum blanca en la tonalitat. Es defineix un espai de color per tal d'estandarditzar la seva representació. En essència, un espai de color es l'especificació d'un sistema de coordenades i un subespai en el qual cada color es veurà representat per un punt.

Al nostre treball usarem l'espai RGB i l'espai HSV que descrivim als següents dos apartats.

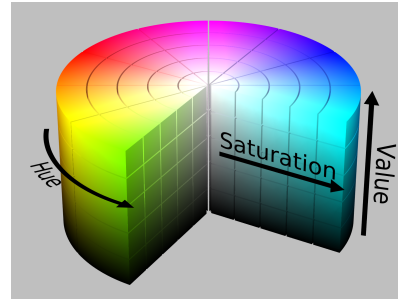
2.3.2 Espai de color RGB

Defineix un color com la quantitat de vermell, verd i blau particular que el formen. Es basa en el sistema de coordenades cartesianes. El subespai d'interès és el que podem veure a la Figura 10c els colors vermell, verd i blau són sobre els eixos de referència i a les altres cantonades tenim el magenta, el cian i el groc. El negre és a l'origen i el blanc a la cantonada més llunyana de l'origen. En aquest model l'escala de grisos forma una línia recta des de negre fins al blanc. [7]

Una manera de definir els colors en aquest cub, que és com ho fa OpenCV, és agafar d'aresta del cub com a 255 i subdividir cadascun dels eixos del cub intervals de manera que un color en RGB es representi com a tres enters.



(a) Cub RGB amb 6 divisions per eix.



(b) Cilindre HSV

Figura 6: Espais de color d'utilitat per al treball

2.3.3 Espai de color HSV

En aquest espai cada tonalitat és a una secció radial al voltant d'un eix central de colors neutrals que van del rang de negres a blancs de baix a dalt. Es representa també amb 3 valors però que indiquen la tonalitat, la saturació i la lluminositat. Veieu Figura 6.

3 Anàlisi

El problema de la detecció de color a les tires es pot dividir en tres parts: trobar la tira a la imatge, corregir el color de la imatge i quantificar el color de la zona d'interès.

3.1 La detecció de la tira de test a l'espai de la imatge

Actualment existeixen multitud de mètodes de classificació i detecció d'objectes en imatges mitjançant l'aplicació de filtres i contorns. El nostre problema però necessita detectar amb molta exactitud la superfície de l'objecte per tal d'aplicar-hi una transformació geomètrica que ens deixi la tira de test paral·lela a la càmera. Per fer-ho s'usaran dos QRs, un a banda i banda de la tira, una de les tasques serà diferenciar quin QR és quin a la imatge (dret o esquerre), una manera de fer-ho seria per exemple clusteritzar els position patterns que hem descrit a la secció anterior 2.2.1 4a .

3.2 La correcció de color de la imatge

La imatge a analitzar pot haver estat presa amb qualsevol dispositiu i qualsevol condició d'il·luminació. Com el nostre objectiu d'anàlisi primordial és el color, és de gran importància realitzar una correcció de color que ens assegurí que els resultats no es veuran esbiaixats per les condicions de la imatge.

3.3 La quantificació de color

Podem quantificar el color pel seu valor RGB, HSV o fer la proporció de les amplituds del pic de resultat respecte el pic de test.

4 Disseny

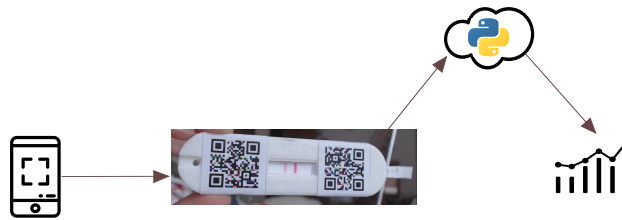


Figura 7: Procès del software

En aquesta secció presentem una descripció d'alt nivell de l'enginy utilitzat per realitzar la tasca de retallat i correcció de color. Les imatges es proveiran per qualsevol tipus de càmera i il·luminació, seran processades per eliminar soroll i els píxels seran tractats com a senyals.

Per tal de poder detectar la zona d'interès i realitzar la correcció de color i de perspectiva es decideix fer ús de QRs (Quick Response Codes) amb una paleta de colors inserida dins.

Els QRs es col·locaran a banda i banda d'un prototip de plàstic blanc amb una obertura a la qual introduir la tira reactiva de manera que, la detecció i retallat de la tira de test es podrà fer amb facilitat trobant els QRs a la imatge i corregint el color gràcies als colors incrustats dins d'ell.

4.1 El prototip on s'introduiran les tires

Trobar quelcom tan singular com una tira biomètrica a l'espai de la imatge pot arribar a ser una àrdua tasca que es pot facilitar enormement si es fa ús d'objectes auxiliars amb patrons de contorns com poden ser QRs. La necessitat de posicionar els QRs respecte a la tira biomètrica a una distància propera i constant ens duu a l'ús d'un prototip de plàstic que contindrà un espai on inserir els QRs i una obertura on s'introduirà la tira. Veure Figura 8.

El color del prototip no és una casualitat, sinó que ha sigut escollit per tal d'introduir el menor error possible en ombres sobre la tira.

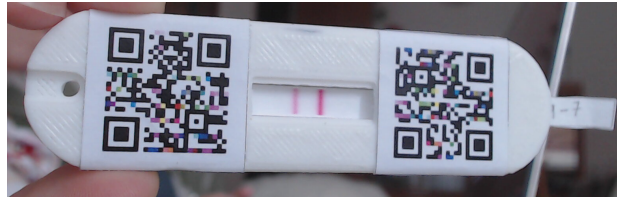


Figura 8: El prototip

4.2 El posicionament dels QRs al prototip

Els QRs es posicionaran al prototip blanc, un girat 180° respecte a l'altre de manera que a cadascuna de les cantonades tinguem un position pattern. Els dos QRs s'hauran de separar i descodificar individualment, una manera de fer-ho pot ser utilitzar un mètode de clusterització dels position patterns com el k-means, ja que sabem que position patterns de QRs diferent sempre estaran separats màximament entre si. L'ús d'aquest mètode permetrà també que la detecció del prototip sigui invariant a l'orientació del prototip.

4.3 Creació de QRs amb color

L'anàlisi de l'estructura i el procés de descodificació a 2.2 ens descobreixen on podem i on no inserir cada color en el nostre generador de QRs amb color i quines condicions hauran de donar-se per tal de preservar les dades contingudes al QR. Tenim doncs:

- Que hem de conservar intactes les zones sensibles del QR, les zones que s'utilitzen per detectar la posició del QR, obtenir informació de la versió, corregir errors... La zona on preferiblement introduïrem els colors és la zona de les dades.
- Que els colors que introduïrem a dita zona han de ser tals que al trobar el llindar a 1 del procés de descodificació que s'explica a 2.2.2 romanguin obscurs o clars depenent de les dades que es desitgi introduir.
- Que la mida idònia per al nostre prototip és la de versió 2, ja que és la versió

més petita que permet alignment patterns i aquests ens seran d'utilitat per millorar l'alineament del QR.

5 Desenvolupament

En aquest apartat s'explicaran els mètodes i tècniques de visió per computador i processament d'imatges utilitzats per tal de dur a terme la detecció i correcció del QR, el retall del prototip en la imatge i la correcció de color. Les eines que s'utilitzaran són OpenCV amb Python i els mòduls de descodificació de QR i correcció de color de l'equip *ColorSensing*.

Començarem per explicar com s'han creat els QRs amb colors que després s'usaran per corregir el color i retallar la zona d'interès. Esquemàticament i a gran escala, el flux del software és:

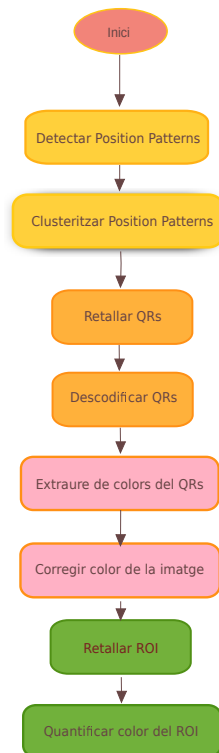


Figura 9: Flux del software

5.1 Correcció de color

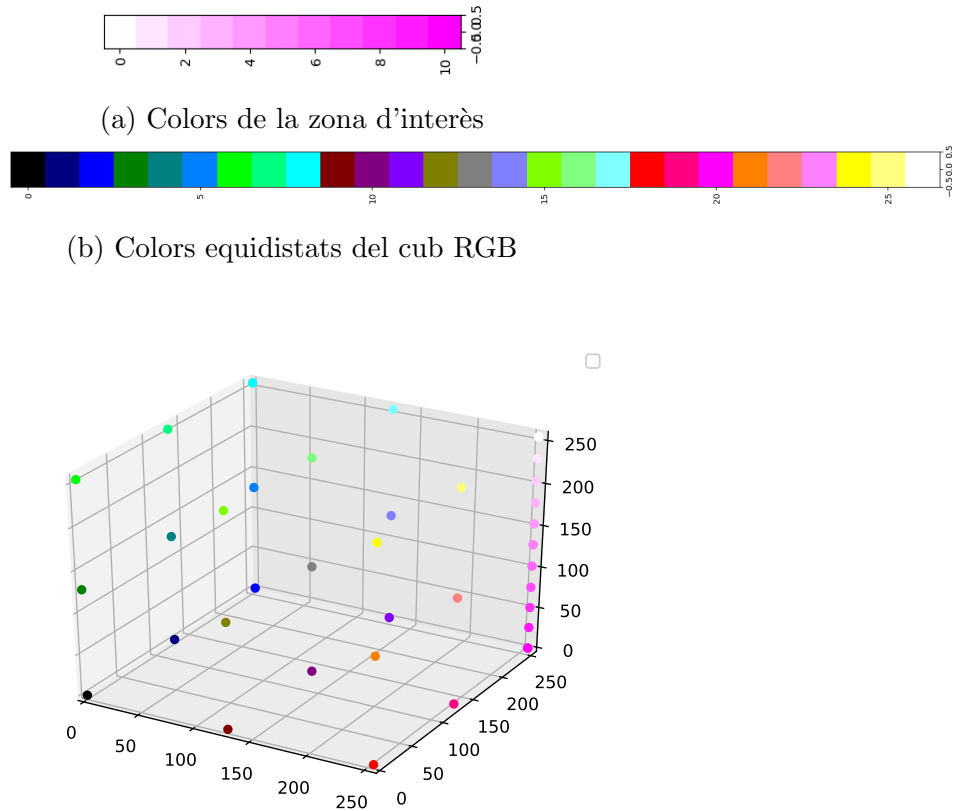
Per tal de realitzar la correcció de color se'ns proporciona el mòdul de correcció de color de l'equip *ColorSensing* que implementa el mètode 3D-TPS per a la correcció de color. [1]

Aquest mètode, a grans trets, mapeja d'un espai RGB a un altre, per tant té com a entrada dos conjunts de colors: l'inici i el destí. Per al nostre cas, el conjunt d'inici correspondrà amb els colors que trobem al QR dins la imatge i el destí el conjunt amb què sabem que hem construït el QR.

5.1.1 Creació de la paleta de colors a inserir

Com a intent per optimitzar la correcció de color els colors que s'han seleccionat per inserir al QR són:

1. Colors equidistants del cub RGB complet per tal de corregir la imatge en general. Figura 10a.
2. Degradat de colors de la gamma que coneixem que pertany al resultat de la tira. Figura 10b.



(c) Colors to insert into the QR in the RGB space in order to correct the general color and the zone of interest

Figura 10: Selecció de colors a inserir als QRs

5.1.2 Creació de QRs amb color

Tal com hem vist a l'apartat del disseny, per construir el nostre QR amb colors per una banda hem de preservar certes zones del QR per no tenir problemes en la detecció i per, altre, al final, el procés de descodificació del QR es podria resumir en aplicar un llindar que separa els mòduls clars dels obscurs que acabaran per convertir-se a bits.

A l'inserir-hi doncs, els colors al QR no volem perdre ni la seva capacitat per ser detectat amb facilitat, ni la informació que conté, és per això que el que es decideix és fer una selecció d'on inserirem els colors a l'espai del QR:

1. Ometrem tota zona encarregada a la detecció del QR.

2. Per cadascun dels colors que vulguem inserir farem la mitjana aritmètica dels seus canals (o cosa que és el mateix, el convertirem a escala de grisos).
3. Si el valor en escala de grisos del color a inserir és inferior a 127, s'inserirà el color a un mòdul de les dades que es vulgui detectar com a obscur, si és superior o igual a 127 s'inserirà el color a un mòdul de les dades que es vulgui detectar com a clar.



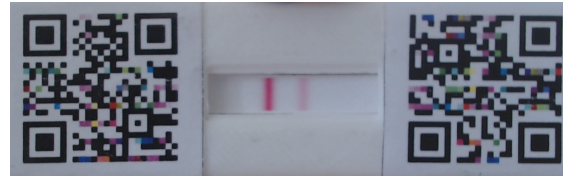
(a) QR típic en blanc i negre.



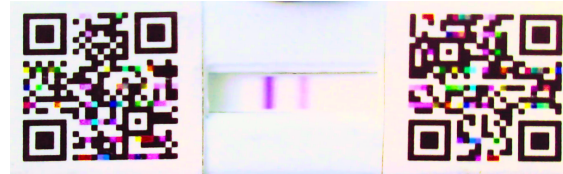
(b) QR amb colors inserits

Figura 11: QR amb i sense color

Una vegada creats els QRs amb color els imprimim i els enganxem al prototip. Modificant el procés de descodificació del QR vist a 2.3 de manera que a banda d'extreure les dades del QR també extregui els colors inserits dins d'ell, podem fer la correcció de color amb el mètode 3D-TPS que com ja hem dit ens proporciona l'equip de *ColorSensing*. Veiem a la Figura 13 la matriu de colors extreta dels QRs i el resultat de la correcció de color a la Figura 12.



(a) Imatge no corregida



(b) Imatge corregida amb el mètode 3D-TPS

Figura 12: Correcció de la imatge amb 3D-TPS i els colors inserits al QR

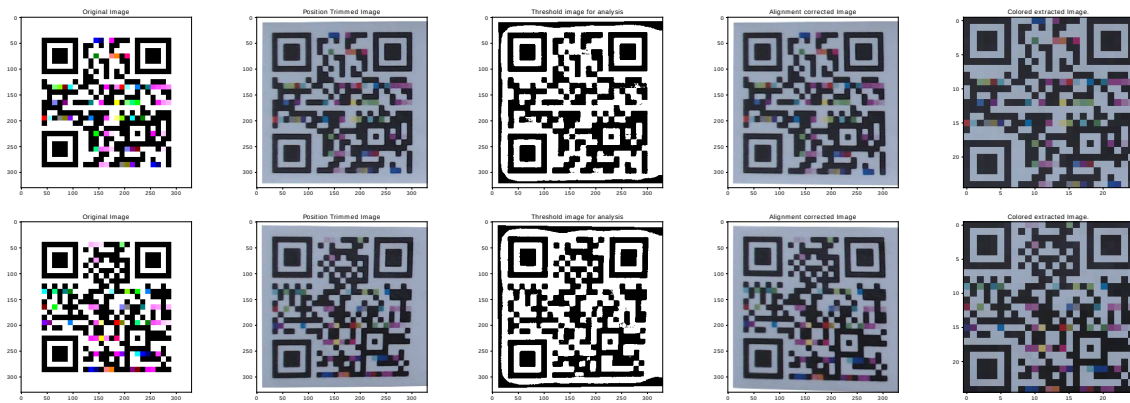


Figura 13: QR creat vs imprès i correcció de l'orientació, perspectiva i matriu de colors extrets de l'imprès

5.2 Detecció del prototip i zona d'interès

Detectar el prototip amb la tira serà tan senzill com detectar el que abans hem vist com positions patterns dels QRs que hem inserit en ell. Com és habitual per tal de detectar objectes els passos a seguir han sigut:

1. **Convertir la imatge en escala de grisos:** amb l'anomenat llindar gaussià adaptatiu. Aquest mètode ens permet que en comptes d'utilitzar un valor global com a llindar es van calculant llindars parcials per cada regió de la imatge, de manera que les condicions d'il·luminació no afectaran negativament. A més a més al ser gaussià ens permetrà reduir el soroll a la imatge, veieu la Figura

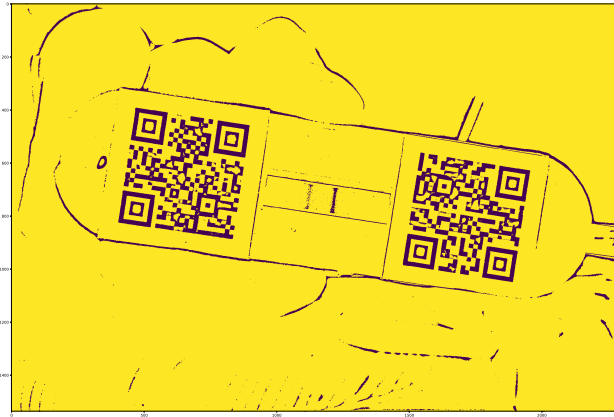
14a.

2. **Aplicar un filtre Canny:** que ens permet detectar vores calculant els gradients de cadascuna de les zones de la imatge, veieu la Figura 14b.
3. **Detectar contorns:** com a col·lecció de vores unides.

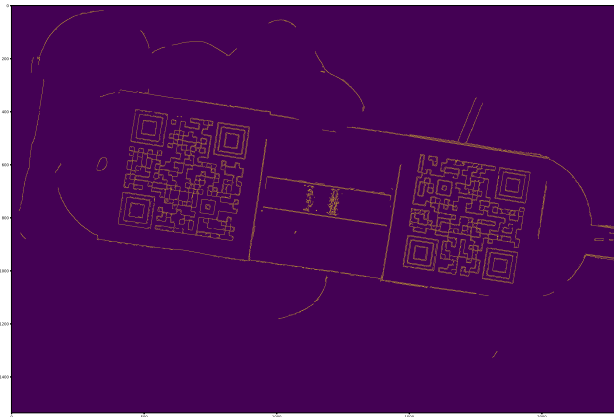
Com a pas més específic per trobar els QRs del prototip.

4. **Buscar la relació 7:5:3** característica dels position patterns vists a la secció 2.2.1 punt 4a.
5. **Clusteritzar els position patterns:** per saber a quin QR dels dos corresponen.
6. **Aplicar els procediments de detecció i correcció de l'orientació:** vists a 2.3 per a cada QR.
7. **Retallar el prototip i la zona d'interès:** gràcies al posicionament dels position patterns i la mesura del costat del QR.

Veurem amb més detall la implementació dels punts 5 i 7 als apartats següents.



(a) Gaussià adaptatiu



(b) Canny

Figura 14: Filtres usats per tal de trobar i descodificar els QRs.

5.2.1 Detecció de múltiples QRs

Abans de trobar l'orientació dels QRs s'haurà d'identificar quins position patterns corresponen a cada QR. Per fer-ho s'usarà el mètode de clusterització k-means que donades n observacions (els n position patterns trobats) les agrupa en k grups (k QRs), veiem el funcionament de l'algoritme com l'explica a [8]

Es tracta d'un algoritme iteratiu que comença amb un conjunt k de centroides (μ). A cadascuna de les actualitzacions, totes les observacions x són assignades al centroide més proper.

$$S_i^{(t)} = \{x_p : \|x_p - \mu_i^{(t)}\|^2 \leq \|x_p - \mu_j^{(t)}\|^2 \forall j, 1 \leq j \leq k\} \quad (5.1)$$

A continuació els centroides són reposicionats en l'espai calculant la mitjana de les observacions observades a dit centroide.

$$\mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \quad (5.2)$$

El procés acaba quan totes les observacions romanen assignades al mateix centroide i per tant aquests ja no canvien més.

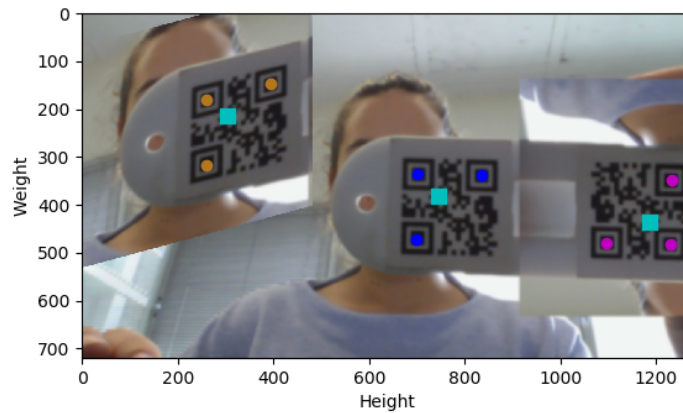


Figura 15: *Position Patterns* clusteritzats, cada color representa un clúster, el quadrat blau cel el centroide del mateix

5.2.2 Retall del prototip

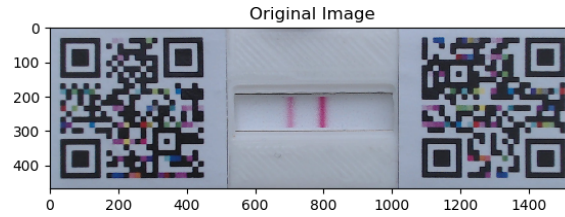
Amb els dos QRs detectats i coneixent la seva orientació podem fer un retall del prototip. Per construcció hem posicionat els QR a banda i banda del prototip de manera que a cada cantonada tinguem un position pattern. Sabent que es correspondran als position pattern dret i superior-dret d'ambdós QRs (suposant la posició natural del QR com a la Figura 11) i que un dels QRs està girat 180° podem agafar els position patterns de les cantonades i aplicar una transformació projectiva i retall del prototip.

5.2.3 Retall de la zona d'interès

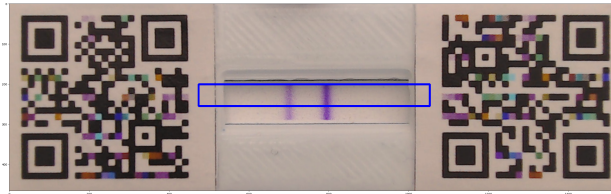
La zona d'interès o ROI (region of interest) donat el prototip es pot aconseguir retallant gràcies al fet que coneixem la mesura del costat del QR i podem compararlo amb la mesura d'aquest en la imatge. Sabent aquesta proporció podrem eliminar les vores del prototip (incloent-hi els QRs) que no formen part de la tira reactiva.



(a) Imatge capturada



(b) Retall i transformació per QRs



(c) Zona d'interès marcada en rectangle blau

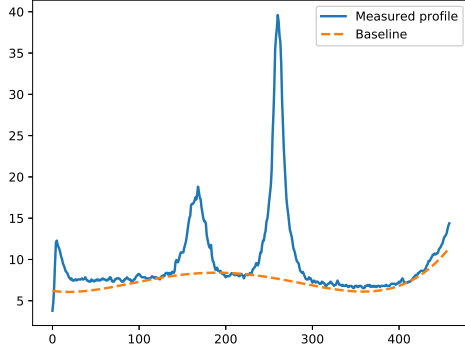
Figura 16: Imatges resultants de cadascun dels passos de detecció i retall

5.3 Quantificació de color

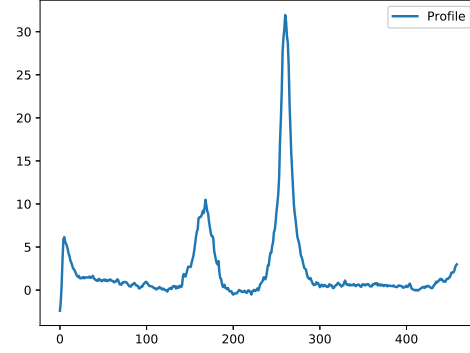
Amb el retall de la ROI obtinguda a 16c ja podem quantificar el color de cadascuna de les ratlles que han donat resultat a la tira. Una manera de fer-ho és calcular la mitjana aritmètica de les columnes de la imatge i fitar el resultat amb un model de pics per tal de trobar les alçades relatives entre les dues campanes.

5.3.1 Extracció de la línia de base

El primer pas per integrar els pics és calcular la línia de base, per fer-ho es fa un ajust polinomial de les zones on no hi ha pics. Veieu la Figura 17b



(a) Línia de base calculada i perfil



(b) Perfil amb línia de base extreta

Figura 17: Càlcul i extracció de la línia de base

5.3.2 Ajust per mínims quadrats no lineals

És una forma d'anàlisi per mínims quadrats per models no lineals amb n paràmetres desconeguts (els paràmetres del model de pic). Principalment el que fa és aproximar el model per un model lineal i anar refinant els paràmetres a cada iteració tal com s'explica a [9]:

Sent m els nostres punts extrets de la ROI amb la línia de base extreta, $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$, el model de pic $y = f(x, \beta)$, que depèn d' n paràmetres, $\beta = (\beta_1, \beta_2, \dots, \beta_n)$, amb $m \geq n$. Volem trobar el vector β de paràmetres que millor s'ajusta a les dades, és a dir aquell que minimitza la suma de quadrats:

$$RSS = \sum_{i=1}^m r_i^2$$

r_i donat per

$$r_i = y_i - f(x_i, \beta)$$

per a $i = 1, 2, \dots, m$. El RSS és mínim quan el gradient és zero:

$$\frac{\partial RSS}{\partial \beta_j} = 2 \sum_i r_i \frac{\partial r_i}{\partial \beta_j} = 0 \quad (j = 1, \dots, n).$$

En un sistema no lineal les derivades $\frac{\partial r_i}{\partial \beta_j}$ són funcions que depenen de les variables independents i dels paràmetres, per tant aquestes equacions no tenen solució única.

El que es fa és seleccionar uns valors inicials pels paràmetres i anar-los aproximant iteració a iteració.

$$\beta_j \approx \beta_j^{k+1} = \beta_j^k + \Delta\beta_j.$$

Aquest problema es pot resoldre amb diferents mètodes numèrics que tenim implementats al paquet científic de Python *lmfit*.

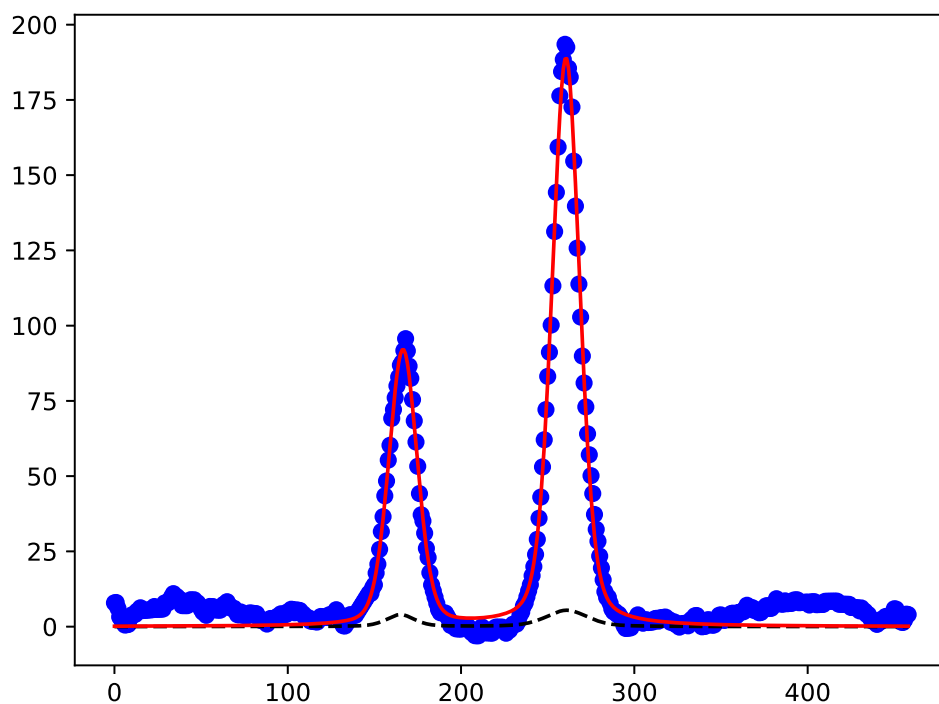


Figura 18: Ajust del perfil per mínims quadrats no lineals, la línia de punts negre pertany a l'ajust amb els valors inicials dels paràmetres, els punts blaus el perfil extret del ROI i la línia vermella l'ajust per a un model de pic PseudoVoigt.

6 Planificació

Com a la majoria de projectes tecnològics plantegem un mètode iteratiu amb els seus corresponents tests. Segons la divisió feta en l'anàlisi del problema fem una predicció de la dificultat de cadascuna de les característiques i plantegem la integració contínua de cadascuna en un software. De major a menor càrrega tenim:

1. Detecció de la zona d'interès
 1. Detecció de QR
 2. Retall del prototip
 3. Retall de la ROI
2. Quantificació del color
 1. Ajust de pics
 2. Creació de dummy tires
 3. Test amb dummy tires
3. Correcció de color
 1. Creació de les paletes de colors
 2. Creació de QR amb colors
 3. Test de la detecció de QR amb color
 4. Implementació amb TPS

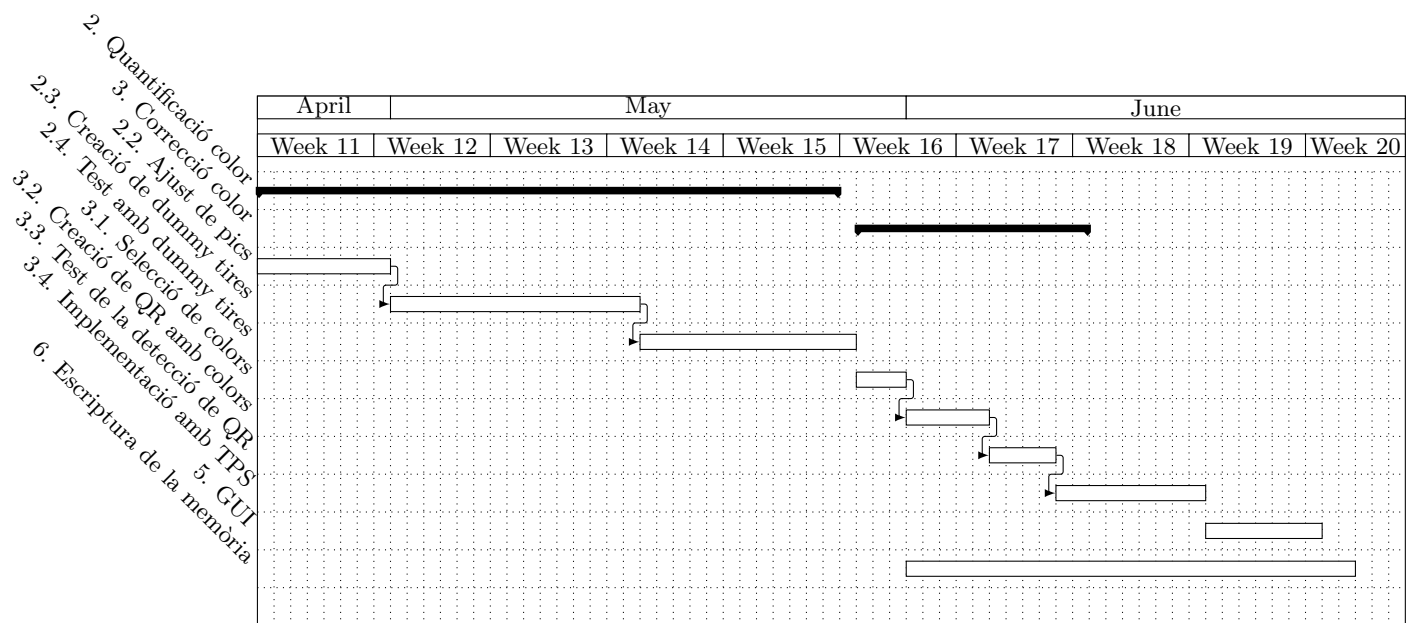
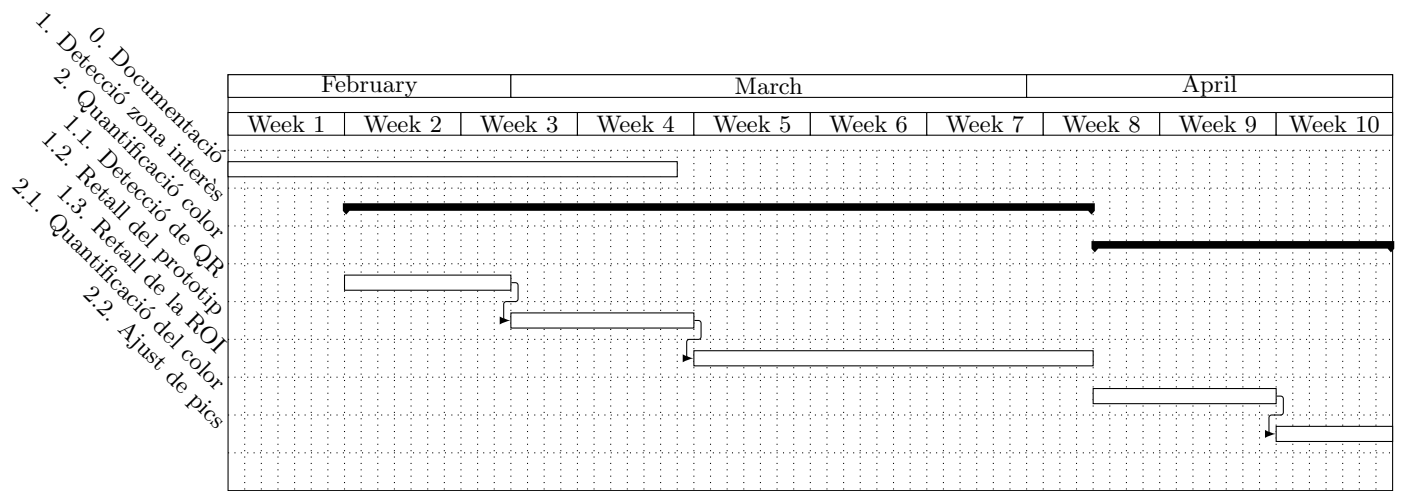


Figura 19: Diagrames de Gantt de la realització del treball

7 Proves i resultats

En aquesta secció s'exposa com s'ha provat el resultat del treball i quin error s'ha obtingut.

7.1 Creació de tires de test artificials

Per provar el software necessitem diferents tires reactives amb models de pics i paràmetres variats. La manera més fàcil i ràpida d'aconseguir-ho és creant artificialment una tira reactiva amb els paràmetres dels pics coneguts.

Tan senzill com construir una imatge amb les dimensions d'una tira reactiva i assignar el valor la suma de dues funcions d'un model de pics al canal de saturació en l'espai HSV de la imatge. Els canals de tonalitat i lluminositat codificats amb la gamma de colors que volem que sigui de la tira.

A l'hora de comparar les alçades dels pics amb els que ens donin de resposta només s'hauran de convertir els colors de l'espai HSV al RGB.

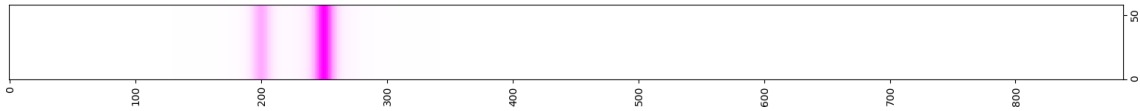


Figura 20: Exemple de tira artificial amb valors de $H = 300$, $V = 100$ i S com a suma de 2 funcions pseudo-voigt amb alçades màximes de 3 i 9 a l'espai HSV.

7.2 Anàlisi dels resultats

A banda de la gràfica de l'ajust que hem vist a la secció 5.3, el software treu per consola un anàlisi més tècnic dels resultats i un resum dels mètodes d'ajust utilitzats de manera que podem tenir en compte l'error. Veiem un exemple d'aquest resum on les variables fitades del primer pic comencen per 'f' i les del segon per 'ss'.

```
[[Model]]  
      (Model(pvoigt, prefix='f') + Model(pvoigt, prefix='ss'))  
[[Fit Statistics]]
```

```

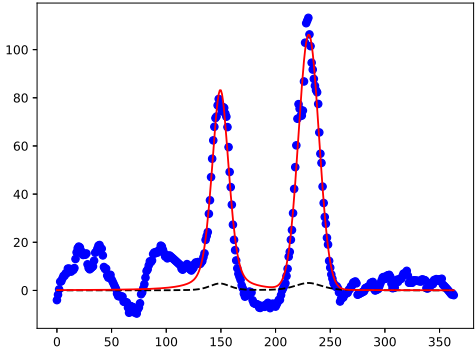
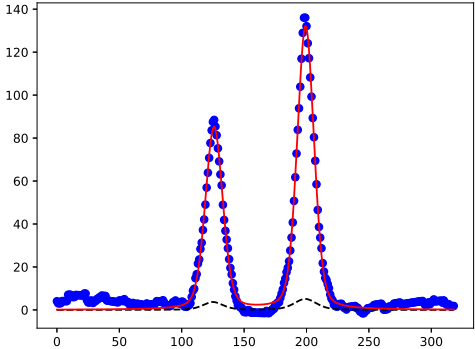
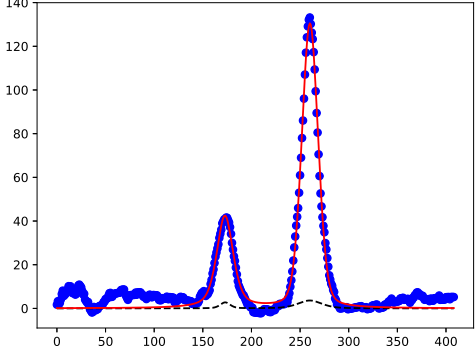
# fitting method    = leastsq
# function evals    = 146
# data points       = 357
# variables         = 9
chi-square          = 5302.84784
reduced chi-square  = 15.2380685
Akaike info crit    = 981.280070
Bayesian info crit  = 1016.17969

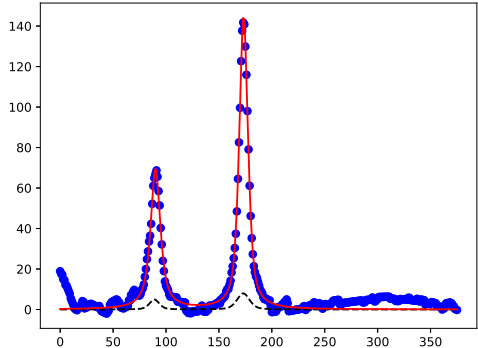
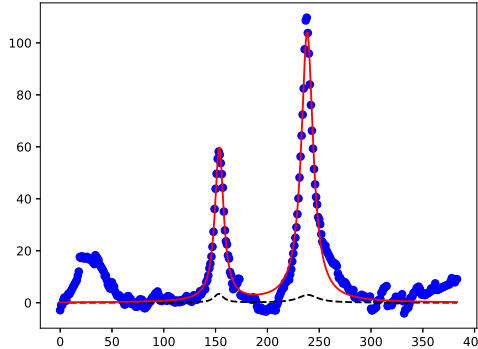
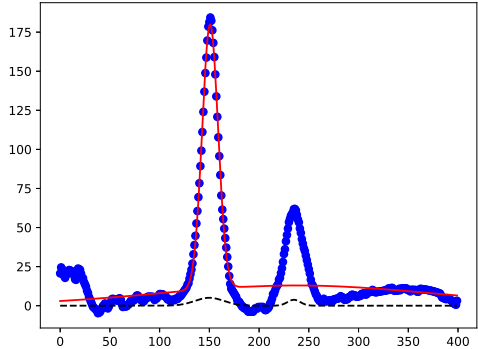
[[Variables]]
ffraction:    0.51098587 +/- 0.00000000 (0.00%) (init = 0.5)
fsigma:       9.64080308 +/- 0.00000000 (0.00%) (init = 8)
fcenter:      120.788495 +/- 0.00000000 (0.00%) (init = 121)
famplitude:   1404.19249 +/- 0.00000000 (0.00%) (init = 60.13017)
fraction:     0.50000000 +/- 0.00000000 (0.00%) (init = 0.5)
fwhm:         19.2816062 +/- 0.00000000 (0.00%) == '2.0000000*fsigma'
fheight:      57.1462967 +/- 0.00000000 (0.00%) == '(((1-ffraction)*famplitude)
ssfraction:   0.26534966 +/- 0.00000000 (0.00%) (init = 0.5)
sssigma:      8.93938277 +/- 0.00000000 (0.00%) (init = 13)
sscenter:     199.596769 +/- 0.00000000 (0.00%) (init = 200)
ssamplitude:  3115.70108 +/- 0.00000000 (0.00%) (init = 149.5079)
ssfwhm:       17.8787655 +/- 0.00000000 (0.00%) == '2.0000000*sssigma'
ssheight:     149.711192 +/- 0.00000000 (0.00%) == '(((1-ssfraction)*ssamplitud

h2/h1=
2.619788170914975

```

Si comparem el resultat que ens retorna el software amb les dades de la construcció de la tira artificial podem arribar a conèixer la mesura d'error del nostre sistema. Veiem una taula 21 algunes d'aquestes comparacions. Agafarem com a referència la proporció en percentatge de quant superior és el pic de test que el de resultat.

Tipus de Peak	Relació original	Relació estreta	Error	Gràfica
PseudoVoigt	128.14%	127%	1%	
PseudoVoigt	132%	155%	15%	
PseudoVoigt	267%	308%	15%	

Tipus de Peak	Relació original	Relació extreta	Error	Gràfica
Lorentziana	159%	208%	31%	
Lorentziana	172%	135%	21%	
Gaussian	200%	253%	26.5%	

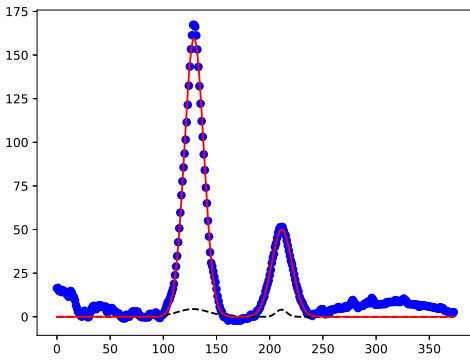
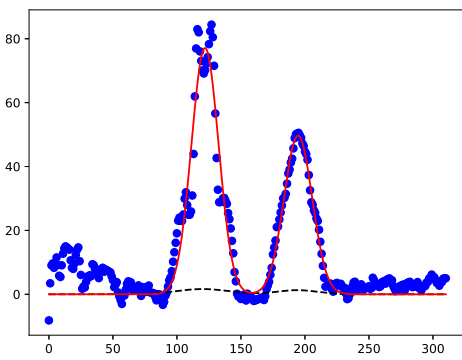
Tipus de Peak	Relació original	Relació estreta	Error	Gràfica
Gaussian	250%	300%	20%	
Gaussian	300%	250%	20%	

Figura 21: Taula de resultats de la relació en percentatges del pic de test amb el de resultat per a diferents tipus d'ajust

Com podem observar l'error del sistema ronda el $\pm 20\text{-}30\%$ a les conclusions comentarem quines podrien ser les causes i com podríem intentar millorar aquest número.

8 Conclusions i treball futur

- Hem aplicat eines de processament d'imatge i visió per computador per trobar els position patterns dels QRs i poder retallar un prototip dissenyat per inserir-hi la tira.
- Hem dissenyat uns QRs amb colors inserits, els hem imprès i els hem utilitzat per realitzar la correcció amb el mètode 3D-TPS.
- Hem quantificat el color retallant la zona dels resultats de la tira i ajustant els pics de color amb funcions gaussianes, lorentzianes i pseudo-voigt.
- Hem provat el nostre sistema fabricant tires artificials, imprimint-les i comparant els resultats obtinguts amb els de fabricació.
- Ens ha quedat un sistema amb un error del $\pm 20\text{-}30\%$, acceptable per al nostre propòsit.

La proposta que considerem per millorar el treball està relacionada a millorar la precisió del sistema. Podríem, a l'elaboració dels QRs amb color, seleccionar colors a corregir més naturals i en més quantitat, de manera que aconseguiríem corregir el color de manera més òptima.

Índex de figures

1	Tires reactives, funcionament i resultats possibles	4
2	Models de pics amb $\mu = 400$, $\sigma = 200$, amplitud = 2, 1.5 i 1.97 respectivament i $\alpha = 0.5$ a la pseudo-voigt	5
3	Estructura d'un <i>Quick Response Code</i>	6
4	La màgia de la detecció de QRs, la relació única dels position patterns	7
5	Metodologia per trobar l'orientació d'un QR.	8
6	Espais de color d'utilitat per al treball	10
7	Procès del software	12
8	El prototip	13
9	Flux del software	15
10	Selecció de colors a inserir als QRs	17
11	QR amb i sense color	18
12	Correcció de la imatge amb 3D-TPS i els colors inserits al QR . . .	19
13	QR creat vs imprès i correcció de l'orientació, perspectiva i matriu de colors extrets de l' imprès	19
14	Filtres usats per tal de trobar i descodificar els QRs.	21
15	<i>Position Patterns</i> clusteritzats, cada color representa un clúster, el quadrat blau cel el centroide del mateix	23
16	Imatges resultants de cadascun dels passos de detecció i retall . . .	24
17	Càlcul i extracció de la línia de base	25
18	Ajust del perfil per mínims quadrats no lineals, la línia de punts negre pertany a l'ajust amb els valors inicials dels paràmetres, els punts blaus el perfil extret del ROI i la línia vermella l'ajust per a un model de pic PseudoVoigt.	26
19	Diagrames de Gantt de la realització del treball	28

20	Exemple de tira artificial amb valors de $H = 300$, $V = 100$ i S com a suma de 2 funcions pseudo-voigt amb alçades màximes de 3 i 9 a l'espai HSV.	29
21	Taula de resultats de la relació en percentatges del pic de test amb el de resultat per a diferents tipus d'ajust	33

Referències

- [1] Benito-Altamirano, I., Casals, O., Fàbrega, C., Waag, A., & Prades, J. D. (2016). *Automated real color calibration using RGB cameras*.
- [2] ISO/IEC 16022:2006: Information technology – Automatic identification and data capture techniques – Data Matrix bar code symbology specification. International Organization for Standardization, Geneva, Switzerland (2006).
- [3] ISO/IEC 16023:2000: Information technology – International symbology specification – MaxiCode. International Organization for Standardization, Geneva, Switzerland (2000).
- [4] ISO/IEC 15438:2015: Information technology – Automatic identification and data capture techniques – PDF417 bar code symbology specification. International Organization for Standardization, Geneva, Switzerland (2015).
- [5] ISO/IEC 18004:2015: Information technology – Automatic identification and data capture techniques – QR Code bar code symbology specification. International Organization for Standardization, Geneva, Switzerland (2006).
- [6] Forsyth, D., & Ponce, J. (2015). *Computer vision: a modern approach*. Boston: Pearson.
- [7] Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing* (3rd ed.). Upper Saddle River, N.J. : Pearson/Prentice Hall.
- [8] MacQueen, J. B. (1966). *Some methods for classification and analysis of multivariate observations*. Ft. Belvoir: Defense Technical Information Center. Retrieved from <https://projecteuclid.org/euclid.bsmmsp/1200512992>
- [9] M.J. Box, D. D., & W.H. Swann. (1969). *Non-Linear optimisation Techniques* Oliver and Boyd for Imperial Chemical Industries, Ed.

- [10] Szeliski, R. (2011). Computer vision: algorithms and applications. London: Springer.